Two-stage Color Texture Synthesis using the Structure Tensor Field

Adib Akl[1,2], Charles Yaacoub[2], Marc Donias[1], Jean-Pierre Da Costa[1] and Christian Germain[1]
[1]*Bordeaux University, IMS Lab, UMR CNRS 5218, Talence cedex, France*
[2]*Faculty of Engineering, Holy Spirit University of Kaslik (USEK), Jounieh, Lebanon*

Keywords: Texture Synthesis, Color Structure Tensor, Color Texture, Multi-scale, Non Parametric Synthesis.

Abstract: Since reproducing the realism of the physical word is a major goal for computer graphics, color texture synthesis is important for rendering synthetic images and animations. Most of the existing synthesis techniques provide impressive results in many cases, but fail in difficult situations with large patterns, or with long range directional variations. Based on a previously developed two-stage structure/texture synthesis algorithm where the structure tensor is used to represent the structure layer, an extension to color texture synthesis is proposed. Two different methods are used for the computation of the color structure tensor field. An acceleration method for the proposed algorithm is also presented. Results show that the proposed approach successfully synthesizes the output texture in many situations where traditional algorithms fail to reproduce the exemplar's patterns and dynamics. These promising results pave the way towards 3D color textures synthesis showing multi-scale patterns.

1 INTRODUCTION

Texture synthesis has been particularly dynamic with different applications in computer vision including image extrapolation, restoration, editing and compression. It has also been extended to video completion/merging and animations, and the description of the geometry of a surface (Bargteil et al., 2006, Yamauchi et al., 2003, Bertalmio et al., 2000, Winkenbach et al., 1994). Most of graphics applications require color texture synthesis to represent real word textures observed under different lighting conditions. In rendering, textures can mimic the surface details of real objects, ranging from varying the surface's color, perturbing the surface normal, to actually deforming the surface geometry. In pen and ink style illustrations, textures can delineate the tone, shade, and pattern of objects. With hand-drawn pictures, most scanned images are of inadequate size and can lead to visible seams or repetitions if they are directly used for texture mapping (Winkenbach et al., 1994).

Several recent 2D texture synthesis algorithms (Portilla & Simoncelli, 2000, Wei & Levoy, 2000, 2003, Paget & Longstaff, 1998, Kwatra et al., 2003, Kopf et al., 2007, Vanhoey et al., 2013, Efros & Freeman, 2001, Han et al., 2006) achieved success in modeling a large panel of textures, including stochastic and structured textures. For instance, a Markov Random Field texture modeling method is proposed in (Paget & Longstaff, 1998). It mathematically captures the visual characteristics of a texture into a unique statistical model that describes the interactions between pixel values. A synthesis algorithm based on copying patch regions from the sample to the output is proposed in (Kwatra et al., 2003). The method uses a graph cut technique to determine the patch region without choosing its size a-priori, in contrast to other existing methods. In (Portilla & Simoncelli, 2000) an over complete complex wavelet transform is used to parameterize the model by a set of statistics, in the frequency domain, corresponding to basic functions at adjacent locations, orientations and scales. The 2D texture synthesis method in (Wei & Levoy, 2000, 2003) models the texture as a realization of a local and stationary random process. The algorithm starts from an input texture and an output image initialized by a white random noise. The texture is synthesized in a scan-line order. The neighborhood of each output pixel is captured and the most similar neighborhood is searched for in the exemplar based on the Euclidian distance. Then the corresponding pixel is copied to the target position in the output texture.

Many of the above 2D synthesis techniques were extended to the 3D environment, i.e. solid texture synthesis. Among such extensions, the non-

Akl A., Yaacoub C., Donias M., Da Costa J. and Germain C..
Two-stage Color Texture Synthesis using the Structure Tensor Field.
DOI: 10.5220/0005358301820188
In *Proceedings of the 10th International Conference on Computer Graphics Theory and Applications* (GRAPP-2015), pages 182-188
ISBN: 978-989-758-087-1

parametric approach of (Kopf et al., 2007), which integrates histogram matching to help the global statistics of the synthesized solid converge towards those of the exemplar. Other parametric (Da Costa & Germain, 2010) and non-parametric (Urs, 2013) 2D/3D extensions have also shown their efficiency in many cases including simulation of atomic structure of materials (Leyssale et al., 2009, 2012).

Most of the existing synthesis algorithms are appropriate for color images by considering the three channels (Red, Green and Blue) of the color texture in the RGB model (Wei & Levoy, 2000, 2003, Kwatra et al., 2003).

For the synthesis of structured anisotropic textures, most of existing approaches tend to produce more regular textures than the exemplar (Kopf et al., 2007, Wei & Levoy, 2000). They are hardly able to reproduce long range orientation variations, dealing badly with non-stationary textures presenting undulating, circular or laminar structures. In this case, the prior synthesis of a geometric layer may help in the synthesis of the texture layer (Peyré, 2009). A two-step synthesis approach consisting in first producing a structure layer from the analysis of the original exemplar, and then using this structure layer to constrain the synthesis of the texture itself, is presented in (Akl et al., 2014). In this method, the structure layer is represented by the structure tensor field which captures the dominant orientations and the degree of local anisotropy in the texture.

Based on this latter algorithm, this paper presents an extension to color texture synthesis using two different approaches for the computation of the structure tensor on color images. The proposed method leads to a same or better quality results than those obtained using a standard approach, with a significant computational load reduction.

The remainder of the paper is organized as follows. The computation of the color structure tensor field is presented in Section 2. The proposed two-stage color texture synthesis algorithm is then described in Section 3. Experimental results are discussed in Section 4, and finally conclusions are drawn in Section 5.

2 THE COLOR STRUCTURE TENSOR FIELD

The structure tensor T of a gray-scale image M is the covariance matrix of the first partial derivatives of M, and built from previously estimated gradient fields $\nabla M = [M_x, M_y]$with $M_x = M*G_x$ and $M_y = M*G_y$ where G_x and G_y are Gaussian derivative kernels, and (*) denotes convolution.

At a point *(x, y)*, the structure tensor *T(x, y)* is symmetric and can be written as:

$$T(x,y) = \begin{bmatrix} T_{xx}(x,y) & T_{xy}(x,y) \\ T_{xy}(x,y) & T_{yy}(x,y) \end{bmatrix}; \tag{1}$$

the tensor components T_{xx}, T_{xy}, and T_{yy} are given by:

$$T_{xx} = S*(M_x M_x); \; T_{xy} = S*(M_x M_y); \; T_{yy} = S*(M_y M_y), \tag{2}$$

where S is a weighting function used for gradient field smoothing.

The structure tensor can be interpreted as an ellipse (Toujas et al., 2010) characterized by a shape (or coherence) indicator and an orientation factor. The former is given by:

$$C(T) = \left(\lambda_1(T) - \lambda_2(T)\right) / \left(\lambda_1(T) + \lambda_2(T)\right), \tag{3}$$

where $\lambda_1(T)$ and $\lambda_2(T)$ are the tensor eigenvalues. The latter is computed from the eigenvector $[e_x, e_y]$ associated with $\lambda_1(T)$ as:

$$\theta(T) = tan^{-1}\left(e_y / e_x\right), \tag{4}$$

The first stage of the texture synthesis algorithm in (Akl et al., 2014) consists of synthesizing the texture's structure layer represented by the structure tensor field. Therefore, an important issue in the color synthesis extension is the computation of the color structure tensor. (Zenzo, 1986) proposes a tensor formulation for the gradient of a multi-component image with the extraction of a single vector (direction and magnitude of maximum variation). (Weijer & Gevers, 2004) propose to add the three structure tensor components computed for each color channel in an RGB image. The same applies in (García et al., 2008) with an additional Gaussian smoothing.

In this paper, two different approaches are used for the computation of a color image tensor-field. The first approach consists of extracting the luminance component (Y) from the color input texture, as defined in the ITU-R BT.601 recommendation (ITU-R, 2011):

$$Y = 0.299\,R + 0.587\,G + 0.114\,B, \tag{5}$$

where R, G, and B represent the Red, Green and Blue components of the input image, respectively. The structure tensor is then computed from the luminance component the same way it is computed from the gray-scale images.

The second method, used in this paper, relies on the additivity of tensors for different channels as (Weijer & Gevers, 2004):

$$T^C(x,y)=\begin{bmatrix}\sqrt{T_{xx}^C} & \sqrt{T_{xy}^C}\\ \sqrt{T_{xy}^C} & \sqrt{T_{yy}^C}\end{bmatrix}; \tag{6}$$

the color tensor components $\sqrt{T_{xx}^C}$, $\sqrt{T_{xy}^C}$ and $\sqrt{T_{yy}^C}$ are expressed as:

$$\sqrt{T_{ab}^C}=\sum_{C=R,G,B} T_{ab}^C,\ ab\in\{xx,xy,yy\} \tag{7}$$

where T_{ab}^R, T_{ab}^G and T_{ab}^B are the structure tensor components obtained on the three channels *R*, *G*, and *B* of the color texture respectively.

3 THE PROPOSED TWO-STAGE COLOR TEXTURE SYNTHESIS

The first stage of the proposed texture synthesis algorithm consists in synthesizing the texture's structure layer represented by the color structure tensor field. Hence, the non-parametric Wei and Levoy (W&L) algorithm (Wei & Levoy, 2000), which usually operates on scalar data, is adapted to the specificities of tensor-valued images.

The algorithm starts by computing the input structure tensor field from the color exemplar. Then an output structure tensor field is initialized by choosing randomly tensors from the input structure tensor field. This field is modified in the synthesis process to look like the input tensor field. Therefore, the neighborhood of the output tensor (a vector of tensors) is first captured, then the most similar neighborhood is searched for in the input tensor field, and the corresponding tensor is copied to the output target position.

The same process is repeated for each output tensor until all the tensors are determined (Akl et al., 2014).

Due to its versatility, the metric in (8) is used to measure the similarity between tensors T_1^C and T_2^C.

$$\chi\left(T_1^C,T_2^C\right)=\left(T_{1xx}^C-T_{2xx}^C\right)^2+\left(T_{1yy}^C-T_{2yy}^C\right)^2+2\left(T_{1xy}^C-T_{2xy}^C\right)^2. \tag{8}$$

The choice of the tensor neighborhood shape and the scan type directly influences the resulting synthesized tensor image. We either use a causal neighborhood with a lexicographical scan type or a square non-causal neighborhood with a completely random walk (Wei & Levoy, 2000).

Since the neighborhood size has to be adequately chosen in order to preserve texture structures, multi-resolution image pyramids can be used to capture the structures more compactly in lower resolution pyramid levels (Wei & Levoy, 2000). This presents an alternative method to the use of large neighborhoods which makes the synthesis computationally expensive while the number of pyramid levels has as much influence as the neighborhood size.

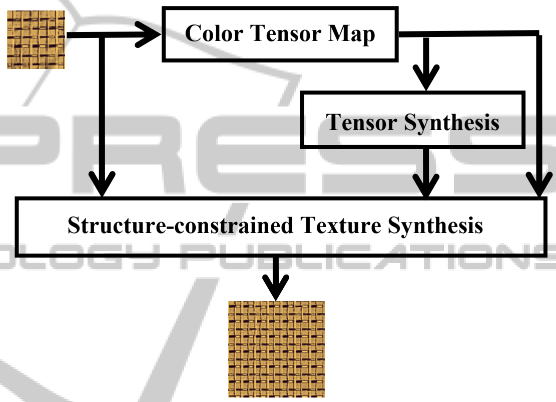

Figure 1: Illustration of the proposed two-stage color texture synthesis algorithm.

The synthesized structure tensor field is used as a constraint for the synthesis. Fig. 1 presents the color texture synthesis method. The algorithm uses as inputs the exemplar, its structure tensor field and the synthesized structure tensor field. The algorithm finds, for every output pixel, the pixel with the most similar neighborhood in the input texture and copies it to the target output position. In other words, each neighborhood has two components: a pixel domain neighborhood in the color texture image, and a tensor domain neighborhood in the structure image (Akl et al., 2014).

The neighborhood resemblance is measured by:

$$R=p\cdot SSCD(N_{in},N_{out})+(1-p)\cdot STD(Q_{in},Q_{out}), \tag{9}$$

where p is a weight factor ($0\le p\le 1$), N is the pixel domain neighborhood component in the input (N_{in}) and output (N_{out}) textures, and Q represents the tensor domain component in the initial (Q_{in}) and synthesized (Q_{out}) tensor field. *SSCD* is the Sum Square Color Distance used for the pixel-domain neighborhood:

$$SSCD(N_1,N_2)=\sum_{n=1}^{N_b}\left[N_1{}^R(n)-N_2{}^R(n)\right]^2 +\left[N_1{}^G(n)-N_2{}^G(n)\right]^2+\left[N_1{}^B(n)-N_2{}^B(n)\right]^2, \tag{10}$$

where N_b is the number of pixels within each neighbourhood and $N_i^C(n),\quad i\in\{1,2\},\quad C\in\{R,G,B\}$ represents the n^{th} pixel within the neighborhood N_i^C. *STD* is the Sum of Tensors Dissimilarity used for the tensor-domain component:

$$STD(Q_1,Q_2)=\sum_{n=1}^{N_b}\chi(Q_1(n),Q_2(n)), \tag{11}$$

where N_b is the number of tensors within each neighborhood and $Q_i(n), i\in\{1,2\}$ represents the n^{th} tensor within the neighborhood Q_i.

In case of multi-scale synthesis, pyramids are obtained by smoothing the tensor field then down-sampling with a 2:1 factor for each additional scale. The synthesis starts from the highest pyramid level and ends at the bottom of the pyramid. To assure that the added high-frequency details are consistent with the already synthesized low-frequency structures, the multi-resolution neighborhood of the current tensor at level *i* contains its same-level neighborhood as well as the neighborhood of the corresponding tensor position at the previously synthesized level (*i*+1).

Due to the additional information provided by the structure map, reproducing the exemplar's patterns is feasible using a smaller texture neighborhood than the one used for the structure layer synthesis (Akl et al., 2014). However, the outperformance of the proposed algorithm still comes at the expense of additional computation of the synthesized structure layer, as will be shown in Section 4. Thus, an algorithm acceleration targeting the structure layer synthesis stage is of our interest.
Practically, the additional computational burden mostly lies in the synthesis of the lowest pyramid level (highest resolution) during a multi-scale synthesis process. In addition, for most of the textures, the structure information in the lowest pyramid level is slightly different than the one existing in the higher level. Therefore, considering a structure tensor pyramid of *L* levels, we propose to synthesize the tensor fields at the high pyramid levels of lower resolution (the coarse levels) up to level *L-1*, and to construct the highest resolution level *L* (the lowest level) from the already synthesized level *L-1* using a bilinear interpolation. This can be used as an interesting alternative to the algorithm in (Akl et al., 2014) which consists in synthesizing at each level of the Gaussian pyramid.

4 RESULTS

This section deals with evaluating the proposed algorithm using different input color textures from Brodatz database (Brodatz, 1966). Due to space limitations, only a subset of the results is presented.

Fig. 2 presents synthesis results on four different color textures. For each result, the first row shows (from left to right) the input texture and the orientation of its structure tensor field computed using the additivity of tensor channels and on the luminance component, the synthesized texture using W&L's algorithm and its orientation image. The second, third and fourth rows show the orientation of the synthesized structure tensor, the resulting texture and its orientation image, respectively, using the additivity of tensors channels for the structure tensor computation (first column), by computing the structure tensor on the luminance component (second column) and using the accelerated algorithm with the structure tensor computation on the luminance component (third column). The best possible parameters are used with W&L and with the proposed algorithm.

Note that no software acceleration (Tree-structured Vector Quantization for example) has been used to overcome any effect related to the sub-optimality of such solutions (Wei & Levoy, 2000).

Three iterations are used to obtain the synthesized structure tensor in result *A*, *B* and *C*, and four iterations in result *D*. For all the results, the textures obtained using the proposed approach are shown after two iterations. For tensor synthesis, two-scale Gaussian pyramids are used in results *A*, *B* and *D* with a neighborhood size of 11×11, 13×13 and 21×21, respectively. Three-scale Gaussian pyramids are used in result *C* with an 11×11 neighborhood size. A mono-scale texture synthesis and a causal neighborhood with a lexicographical scan are used for all the results. The texture neighborhood size is 9×9 in results *A* and *B*, 11×11 and 21×21 in results *C* and *D*, respectively. The palette used for orientation images is shown in the upper center of Fig. 2.

It can be observed that the textures obtained with the tensor-constrained synthesis, using both approaches for color structure tensor computation in results *A* and *B*, are similar to those obtained with W&L. On the contrary, both approaches outperform W&L in results *C* and *D* mainly in structure conservation. With W&L, the synthesized image in result *C* is of acceptable quality, however it is more regular than the exemplar. In result *D*, the texture obtained with W&L presents some undesired artifacts and the alternation and periodicity of the exemplar's patterns are not respected. On the contrary, our approach leads to smooth and artefact-free textures very similar to the input sample giving the impression that they were produced by the same process.

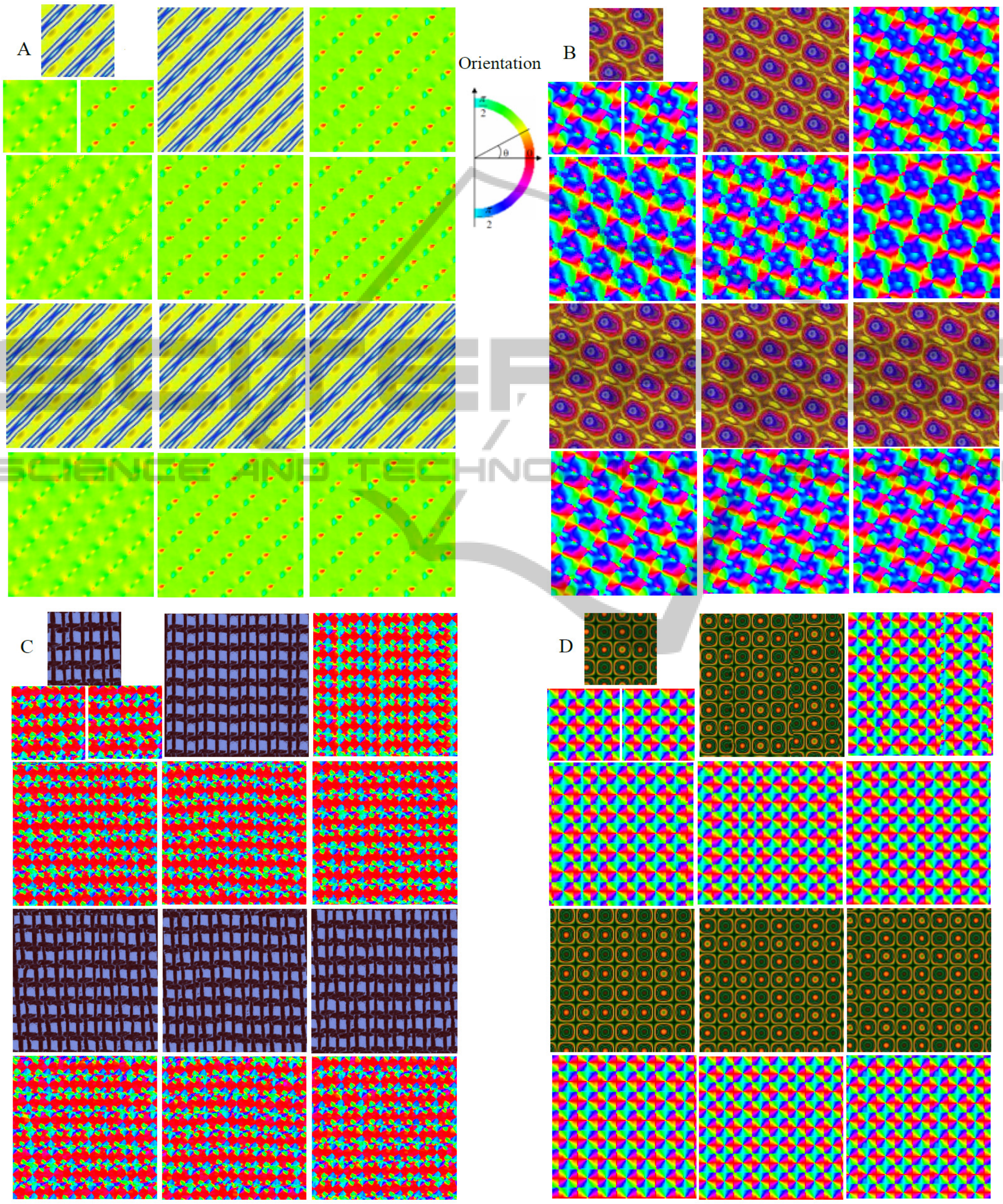

Figure 2: Synthesis results. For each result (A, B, C and D), the 1st row shows (from left to right) the input texture and the orientation of its structure tensor field computed using the additivity of tensor channels and on the luminance component, the synthesized texture using W&L's algorithm and its orientation image. The 2nd, 3rd and 4th rows show the orientation of the synthesized structure tensor, the resulting texture and its orientation image, respectively, using the additivity of tensors channels for the structure tensor computation (1st column), by computing the structure tensor on the luminance component (2nd column) and using the accelerated algorithm with the structure tensor computation on the luminance component (3rd column). The palette used for orientation images is shown in the upper center.

In other words, due to the additional information provided by the structure layer, the proposed approach is able to successfully reproduce the variations of orientations in the exemplar even when W&L fails to sustain the structure.

In results *B*, *C* and *D*, the orientation images of the input tensor fields are almost similar using both approaches for color structure tensor computation, which leads to nearly similar synthesized tensor fields. This is not the case for texture *A* where each approach gives a visually different orientation image. However, synthetic textures obtained with the four different exemplars, using both approaches for color tensor computation, hardly differentiate from each other, leading to eye-friendly synthesized textures of satisfying quality.

It is clearly seen that in all the results, the images obtained using the bilinear interpolation for the structure tensor synthesis stage are roughly similar to those generated by the unaccelerated approach. The interpolated structure tensor remains of good quality and the synthetic orientation images resembles the texture orientation images in terms of both structure and dynamics preservation.

Table 1 presents the simulation time of the results in Fig. 2. The second row shows the running time of W&L's algorithm. The third and fourth rows show the running time of our algorithm using the classical all-levels synthesis method and using the accelerated algorithm by bilinear interpolation, respectively. All the presented timings are in seconds and measured using an Intel Core i7-2670QM CPU with a 2.20 GHz clock.

Table 1: Running time (seconds) for the textures in Fig. 2.

Texture:	A	B	C	D
W&L	28	35	860	756
Proposed Method	41	42	1398	1114
Accelerated Algorithm	20	21	577	424

Unlike the unaccelerated two-stage synthesis method, the proposed accelerated algorithm using the structure tensor interpolation outperforms W&L in terms of time consumption. For example, W&L and the proposed classical approach took 860 seconds and 1398 seconds to generate the output texture in result *C* respectively, while the accelerated algorithm requires 577 seconds.

It is important to mention that even when the pixel-based synthesis algorithm of W&L is able to successfully synthesize the input texture, the accelerated version of the proposed two-stage synthesis is beneficial in simulation time reduction without any loss in the output texture quality, as it is the case in results *A* and *B*.

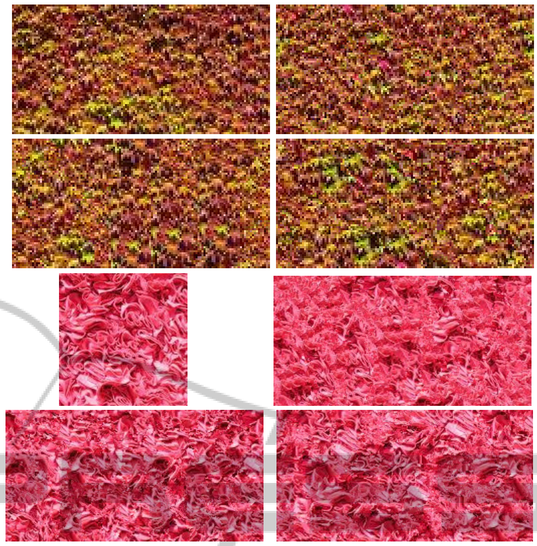

Figure 3: Synthesis results on irregular textures. For each result from top-left to bottom-right; the exemplar, the obtained texture using W&L, the synthesised textures with the proposed unaccelerated method and using the accelerated algorithm.

Fig. 3 shows two synthesis results on irregular long range directional variations textures. Each result shows from top-left to bottom-right; the input texture, the synthesized texture using W&L's algorithm, the obtained results with the proposed unaccelerated and accelerated algorithms.

It can be observed that the textures obtained with the unaccelerated as well as the accelerated tensor-constrained synthesis approaches, in the first result, are similar to those obtained with W&L. On the other hand, W&L fails to reproduce the exemplar's variation of orientations, in the second result, while the textures obtained by the proposed tensor-constrained synthesis are more realistic, better respecting the orientations of the sample structures.

Let us recall that for all the results, the acceleration by Tree-structured Vector Quantization (TSVQ) is not used with W&L's algorithm (Wei & Levoy, 2000) and could be used as well for the proposed structure/texture synthesis method. Thus, future adaptation of such acceleration algorithms to the tensors neighborhood, is of our interest.

5 CONCLUSIONS

This paper presented a non-parametric color texture synthesis algorithm based on two-stage structure/texture processing. The structure layer, represented by the color structure tensor field, is synthesized in the first stage and used as a constraint for texture synthesis in the second stage. Two different methods for color structure tensor computation were developed. An acceleration technique for the proposed algorithm is also presented. The obtained results are highly encouraging, in terms of structure and dynamics preservation, and proved that the proposed method is advantageous for simulation time consumption and for accurately reproducing the exemplar's variations of orientations even when traditional algorithms fail to reproduce the exemplar's patterns. As for future work, we aim at reinforcing the use of the tensor constraint for the synthesis of anisotropic and non-stationary textures and to develop a 2D/3D color texture synthesis algorithm.

REFERENCES

Bargteil, A. W., Sin, F., Michaels, J. E., Goktekin, T. G., O'Brien, J. F., 2006. "A Texture Synthesis Method for Liquid Animations," *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2006*.

Yamauchi, H., Haber, J., Seidel, H-P., 2003. "Image restoration using multiresolution texture synthesis and image inpainting," *Proc. Int. Conf. Comput. Graph.*

Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C., 2000. "Image inpainting," *Proc. of the 27th annual conference on Computer graphics and interactive techniques*, pp. 417-424.

Winkenbach, G., Salesin, D. H., 1994. "Computer-generated pen-and-ink illustration," *Proc. of SIGGRAPH 94*, pp. 91–100.

Portilla, J., Simoncelli, E.P., 2000. "A Parametric Texture Model based on Joint Statistics of Complex Wavelet Coefficients," *Int'l Journal of Computer Vision,* vol.40(1), pp. 49-71.

Wei, L.-Y., Levoy, M., 2000. "Fast texture synthesis using tree-structured vector quantization," *Proc. of ACM SIGGRAPH 2000*, pp. 479-488.

Wei, L.-Y., Levoy, M., 2003. "Texture synthesis from multiple sources," *Proc. of ACM SIGGRAPH Sketches & Application.*

Paget, R., Longstaff, I.D., 1998. "Texture synthesis via a non causal nonparametric multiscale markov random field," *IEEE Trans. on Image Processing*, vol. 7(6), pp. 925–931.

Kwatra, V., Schödl, A., Essa, I., Turk, G., Bobick, A., 2003. "Graphcut Textures: Image and Video Synthesis Using Graph Cuts," *Proc. of ACM SIGGRAPH*, pp. 277-286.

Kopf, J., Fu, C.W., Cohen-Or, D., Deussenn, O., Lischinski, D., Wong, T. T. 2007. "Solid Texture Synthesis from 2D Exemplars," *Proc. of ACM SIGGRAPH*, vol. 26(3).

Vanhoey, K., Sauvage, B., Larue, F., Dischler, J-M., 2013. "On-the-fly multi-scale infinite texturing from example," *ACM Trans. Graph,* 32(6): 208.

Efros, A., Freeman, W. T., 2001. "Image quilting for texture synthesis and transfer," *In SIGGRAPH: Proc. of the 28th annual conference on Computer graphics and interactive techniques,* pp. 341-346.

Han, J., Zhou, K., Wei, L.-Y., Gong, M., Bao, H., Zhang, X., Guo, B., 2006. "Fast example-based surface texture synthesis via discrete optimization," *Visual Computer,* 22(9), pp. 918-925.

Da Costa, J.-P., Germain, C., 2010. "Synthesis of solid textures based on a 2D example: application to the synthesis of 3D carbon structures observed by transmission electronic microscopy," *Proc. of SPIE, Image Processing: Machine Vision Applications III*, vol. 7538, pp. 10.

Urs, R. D., 2013. "Non-parametric synthesis of volumetric textures from a 2D sample", *PhD Thesis, Univ. Bordeaux I.*

Leyssale, J.-M., Da Costa, J.-P., Germain, C., Weisbecker, P., Vignoles, G., 2009. "An image guided atomistic reconstruction of pyrolytic carbons," *App Phys Lett.*, vol. 95(23), pp. 231912.

Leyssale, J.-M., Da Costa, J.-P. Germain, C., Weisbecker, P., Vignoles, G., 2012. "Structural features of pyrocarbon atomistic models constructed from transmission electron microscopy images," *Carbon*, vol. 50(12), pp. 4388-4400.

Peyré, G., 2009. "Texture Synthesis with grouplets," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32(4), pp. 733-746.

Akl, A., Yaacoub, C., Donias, M., Da Costa, J.-P., Germain, C., 2014. "Structure Tensor Based Synthesis of Directional Textures for Virtual Material Design," *Proc of ICIP 2014*.

Toujas, V., Donias, M., Berthoumieu, Y., 2010. "Structure Tensor Field Regularization Based on Geometric Features," *Proc of EUSIPCO 2010*.

Zenzo, S. D., 1986. "A note on the gradient of a multi-image ," *Int'l J. Computer Vision, Graphics Image Process.*, vol. 33, pp. 116-125.

Weijer, J., Gevers, T., 2004. "Tensor Based Feature Detection for Color Images," *12th Color Imaging Conference*: Color Science and Engineering Systems, Technologies, Applications, pp. 100-105.

García, R., Deriche, R., Alberola-López, C., 2008. "Texture and color segmentation based on the combined use of the structure tensor and the image components," *Signal Processing*, vol. 88, pp. 776-795.

ITU-R Recommendation BT.601-7, "Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios," ITU-R, Mar. 2011.

Brodatz, P., 1966. A Photographic Album for Artists and Designers. *Dover Ed., New York.*